

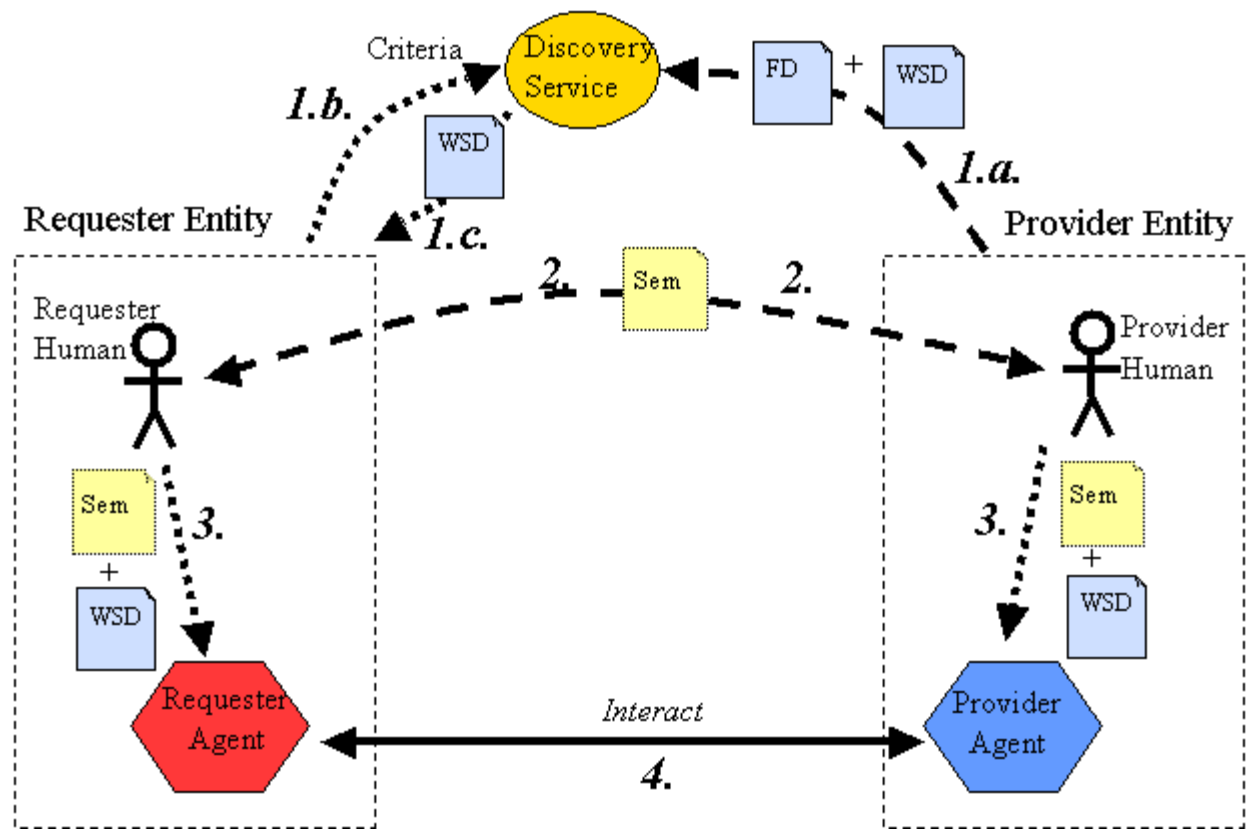
Architectures Orientées Services

Les expressions de processus d'orchestration
BPEL

Sommaire du module

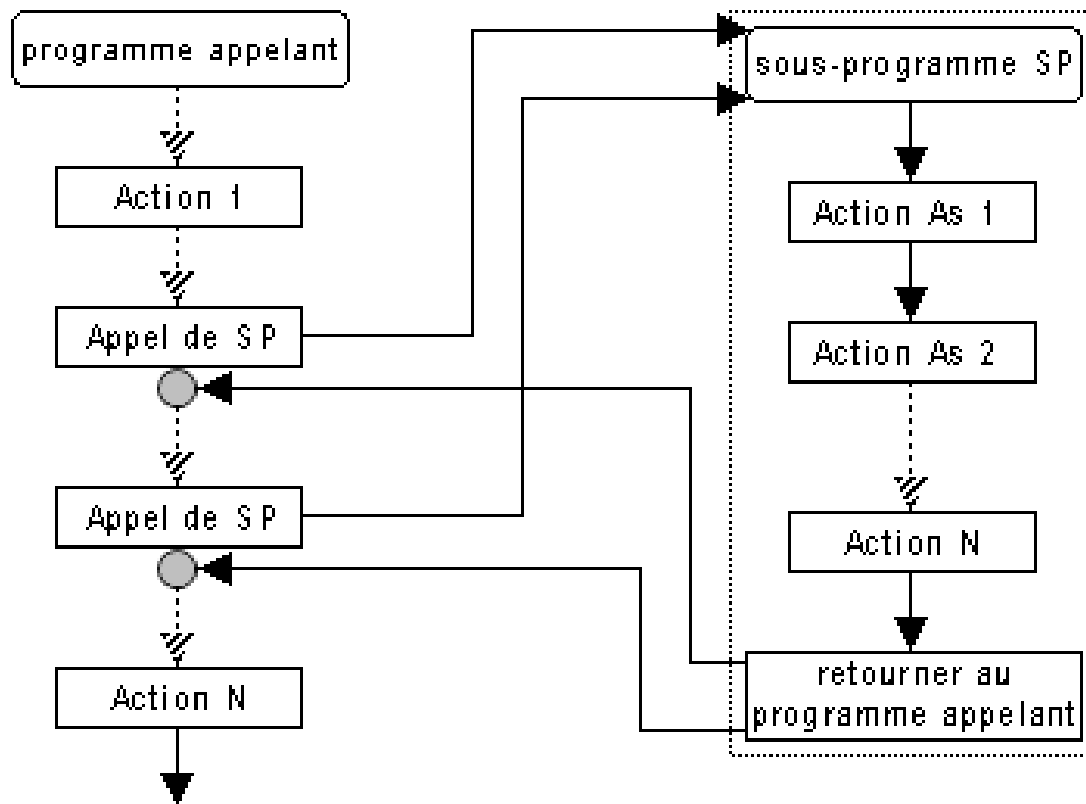
- La problématique de « l'orchestration » des services sur le Web
 - Les langages d'expression de processus
 - BPEL4WS
 - BPEL4People
 - BPMN
- L'initiative des composants standards d'architecture des services SCA

Retour sur les usages de Web services...

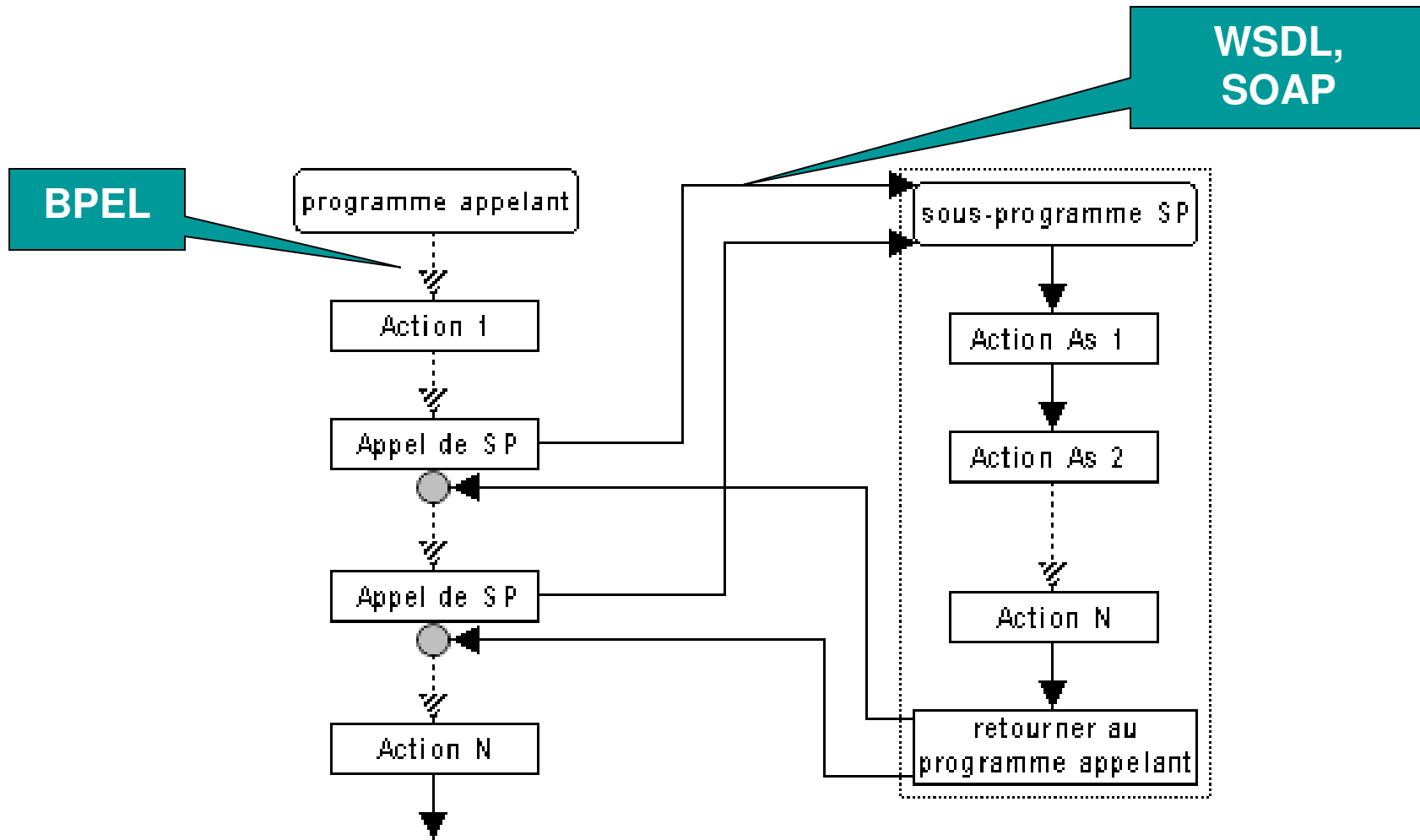


Problématique de l'orchestration des web services

- Une problématique des plus ancienne...



Problématique de l'orchestration des web services

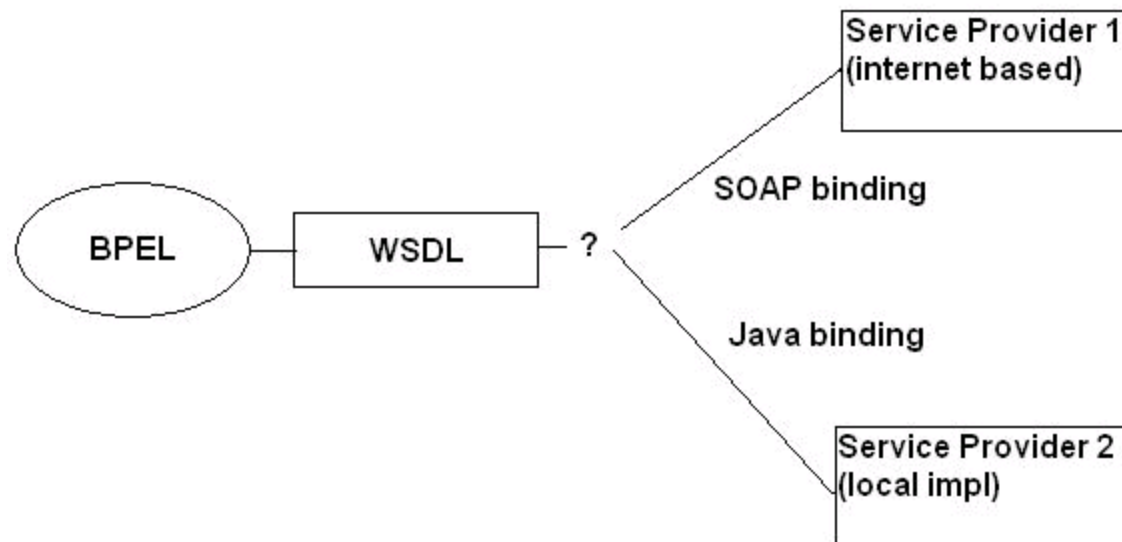


BPEL : Business Process Execution Language

- BPEL est un langage XML standard pour l'expression de procédures d'appel à des services.
- Il se décline en deux standards complémentaires :
 - BPEL4WebServices
 - Définit le langage pour l'organisation des appels aux web services.
 - BPEL4People
 - Définit le langage pour spécifier l'organisation des gestes des acteurs humains dans une architecture de web services.

Rappel sur la notion de services

- Un « web service » n'est que la *description* d'un service ; il ne spécifie en rien où et comment il fonctionne.
- Pour qu'il soit utilisable la description d'un web service doit spécifier un « binding » qui définit l'implémentation de l'appel à ce service.
Le « binding » le plus fréquent est SOAP/HTTP pour appeler des services via le Web ; une autre forme de « binding » est celui de Java qui permet de définir des implémentations locales d'appel à des services.
- Par exemple si vous voulez écrire un web service permettant à des clients d'écrire quelque chose d'original, comme « hello world ! » vous pouvez l'écrire en java, et l'exposer comme un web service.



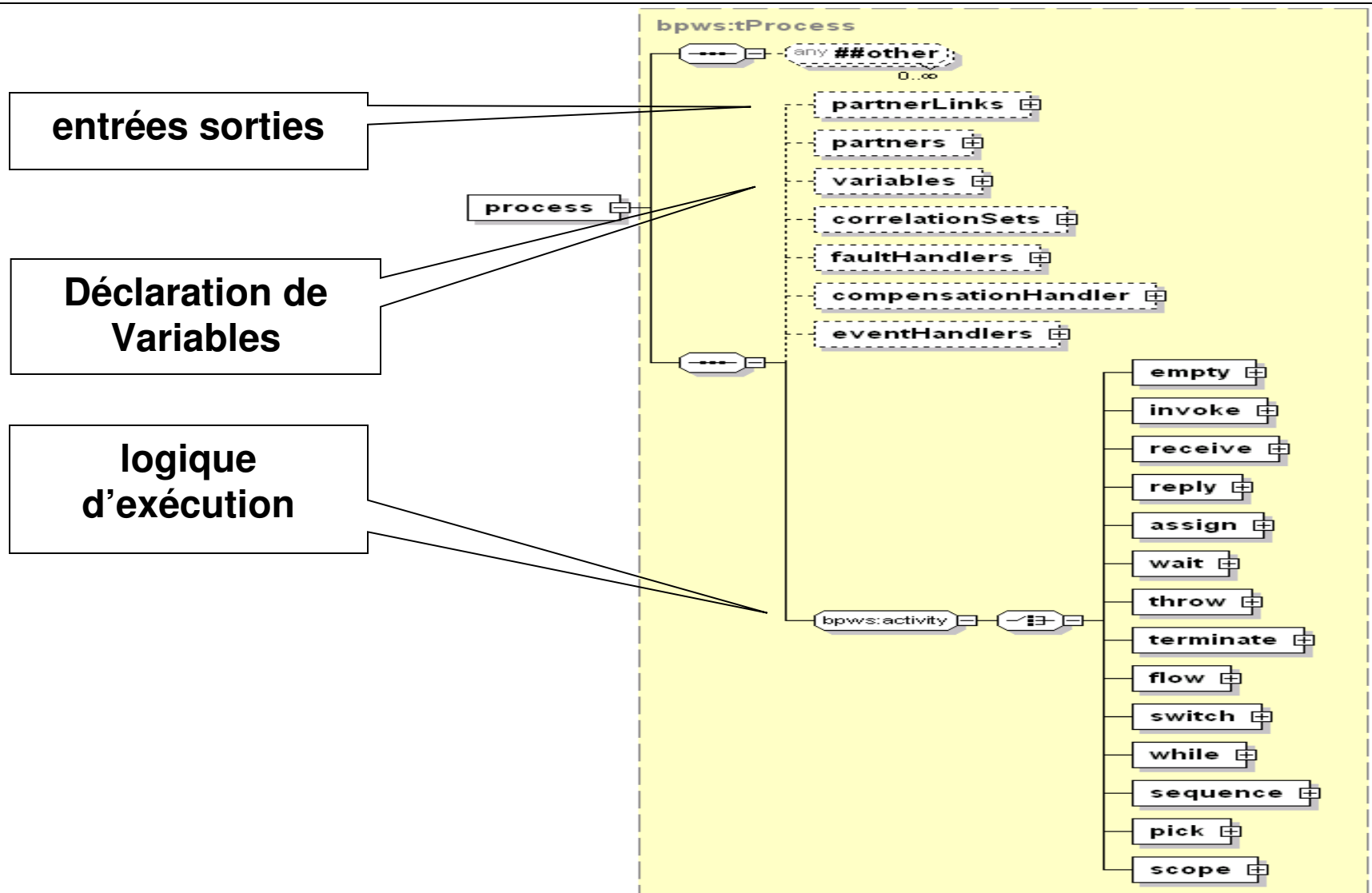
Option d'architecture

- Ordinairement BPEL est un langage de programmation coté serveur:
 - déployé pour servir et suivre une requête venue coté client
 - Déployé comme une sorte de Proxy commode pour un ensemble d'applications anciennes
- Dans les deux cas l'enchaînement typique des événements est:
 - Le web service client demande quelque chose au serveur, via un navigateur ou un autre serveur
 - La requête du client est reçue par le serveur de processus BPEL
 - La requête du client est identifiée comme une nouvelle instance du processus BPEL invoqué pour le servir
 - Le client continue à interagir avec le processus en cours avec des échanges mutuels de messages selon l'organisation du processus, jusqu'à achèvement de celui-ci
 - L'instance du processus disparaît alors que le client achève son activité

BPEL : un langage XML de programmation

- Comme tout langage BPEL comporte :
 - Des déclaration de variables (typées)
 - Utilisant le typage de données de XML-Schema (XSD)
 - Des entrées sorties
 - Faisant appel à des descriptions de web services WSDL
 - Une logique d'exécution
 - BPEL soi même
- L'exemple didactique qui suit montre un appel élémentaire pour un web service aussi élémentaire

La structure d'une expression BPEL



Generated with XMLSpy Schema Editor www.xmlspy.com

Approche par l'exemple : « hello world »

```
<?xml version="1.0" encoding="UTF-8"?>
<process
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:print="http://www.eclipse.org/tptp/choreography/2004/engine/Print"

  <import importType="http://schemas.xmlsoap.org/wsdl/" location="http://blah.wsdl"
    namespace="http://www.eclipse.org/tptp/choreography/2004/engine/Print" />

  <partnerLinks>
    <partnerLink name="printService"
      partnerLinkType="print:printLink"
      partnerRole="printService"/>
  </partnerLinks>

  <variables>
    <variable name="MaVariable" messageType="print:PrintMessage" />
  </variables>
  <assign>
    <copy>
      <from><literal>Hello World</literal></from>
      <to>$MaVariable.value</to>
    </copy>
  </assign>
  <invoke partnerLink="printService" operation="print" inputVariable="MaVariable"/>
</process>
```

Approche par l'exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<process
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process"
  xmlns:print="http://www.eclipse.org/tptp/choreography/2004/engine/Print"

  <import importType="http://schemas.xmlsoap.org/wsdl/" location="http://blah.wsdl"
    namespace="http://www.eclipse.org/tptp/choreography/2004/engine/Print" />

  <partnerLinks>
    <partnerLink name="printService"
      partnerLinkType="print:printLink"
      partnerRole="printService"/>
  </partnerLinks>

  <variables>
    <variable name="MaVariable" />
  </variables>
  <assign>
    <copy>
      <from><literal>Hello World</literal>
      <to>$MaVariable.value</to>
    </copy>
  </assign>
  <invoke partnerLink="printService" operation="print" inputVariable="MaVariable"/>
</process>
```

L'adresse du
web service
invoqué

Les **PartnerLinks** peuvent être vus comme les emplacements pour les objets auxquels vous vous adressez effectivement
Si un web service est décrit par une ressource WSDL, un **PartnerLink** définit une instance du web service invoqué. Il se mappe vers un « **portType** » WSDL de sorte qu'à un partnerLink correspond un seul web service.

Approche

La description d'un web service est semblable à la description de la façon dont on procède pour se faire livrer une pizza : vous appelez, donnez votre adresse, et spécifiez la ou les sortes de pizzas que vous voulez.

L'élément « Partnerlink » est semblable aux Post-It ® sur lequel vous avez reporté le noms et les n° de téléphone de différents marchands de pizza... vous vous servez d'un de ces Post-It pour appeler un marchand particulier (sorte de Web Service). au numéro appelé correspond l'élément d'adresse 'endpoint ' défini dans la description WSDL ou dans la spécification BPEL, où il peut être placé dans un élément partnerLink pour appeler un autre fournisseur.

```
<?xml versi
<process
  xmlns="
  xmlns:
<imp
  nam
```

```
<partnerLinks>
  <partnerLink name="printService"
    partnerLinkType="print:printLink"
    partnerRole="printService"/>
</partnerLinks>

<variable
  <variable name="MaVariable" messageType="print:PrintMessage" />
</variables>
<assign>
  <copy>
```

Un « partnerlinks » spécifie aussi la façon dont d'autres web services s'adressent à vous. Le partner link définit ici le rôle du web service appelé. De façon symétrique, cette définition du « partnerlink » pourrait avoir un attribut 'myRole' qui définirait le web service que cette expression BPEL implémenterait.

```
</as
<invoke partnerLink="printService" operation="print" inputVariable="MaVariable"
</process>
```

Approche par l'exemple

```
<?xml version="1.0"
<process
  xmlns="http://schemas.xmlsoap.org/wsdl"
  xmlns:print="http://schemas.xmlsoap.org/wsdl/soap"
  <import importType="http://schemas.xmlsoap.org/wsdl/soap"
    namespace="http://schemas.xmlsoap.org/wsdl/soap" />
  <partnerLinks>
    <partnerLink name="PrintMessage"
      partnerLinkType="print:PrintMessage"
      partnerRole="client"
      service="PrintMessage" />
  </partnerLinks>
  <variables>
    <variable name="MaVariable" messageType="print:PrintMessage" />
  </variables>
  <assign>
    <copy>
      <from>print:PrintMessage
      <to>$MaVariable
    </copy>
  </assign>
  <invoke partnerLink="PrintMessage" />
</process>
```

Les Variables sont traitées in BPEL soit au travers des « endpoints » des web service ou par assignation. Cet exemple montre l'assignation d'une valeur littérale à la variable nommée 'MaVariable'. Dans ce cas cette variable est un message WSDL avec une partie nommée 'value' de type 'xsd:string' type. Il peut donc il y avoir d'autres contenus de type 'xsd:string' assignée à cette variable.

La syntaxe '\$varname' utilisée pour référencer la variable obéit au standard XPATH. Le séparateur '.' est utilisé pour spécifier la partie WSDL du message . Si cette variable ou cette partie était un complexType XSD un séparateur '/' pourrait être utilisé pour spécifier un sous élément (e.g. '\$MaVariable.value/subvalue').

Les Variables contiennent des données en BPEL. peuvent contenir des valeurs selon un type XSD ou un message WSDL. Dans cet exemple, la variable nommée 'MaVariable' est déclarée comme un conteneur de messages WSDL de type 'print:PrintMessage'. Au lieu d'un attribut 'messageType', la variable aurait pu avoir un attribut 'type' qui aurait défini un type XSD simple ou complexe tel que 'xsd:string' ou 'xsd:integer'.

Les Variables sont utilisées pour passer des paramètres en entrée et en sortie des « endpoints » de web service.



.wsdl
/>

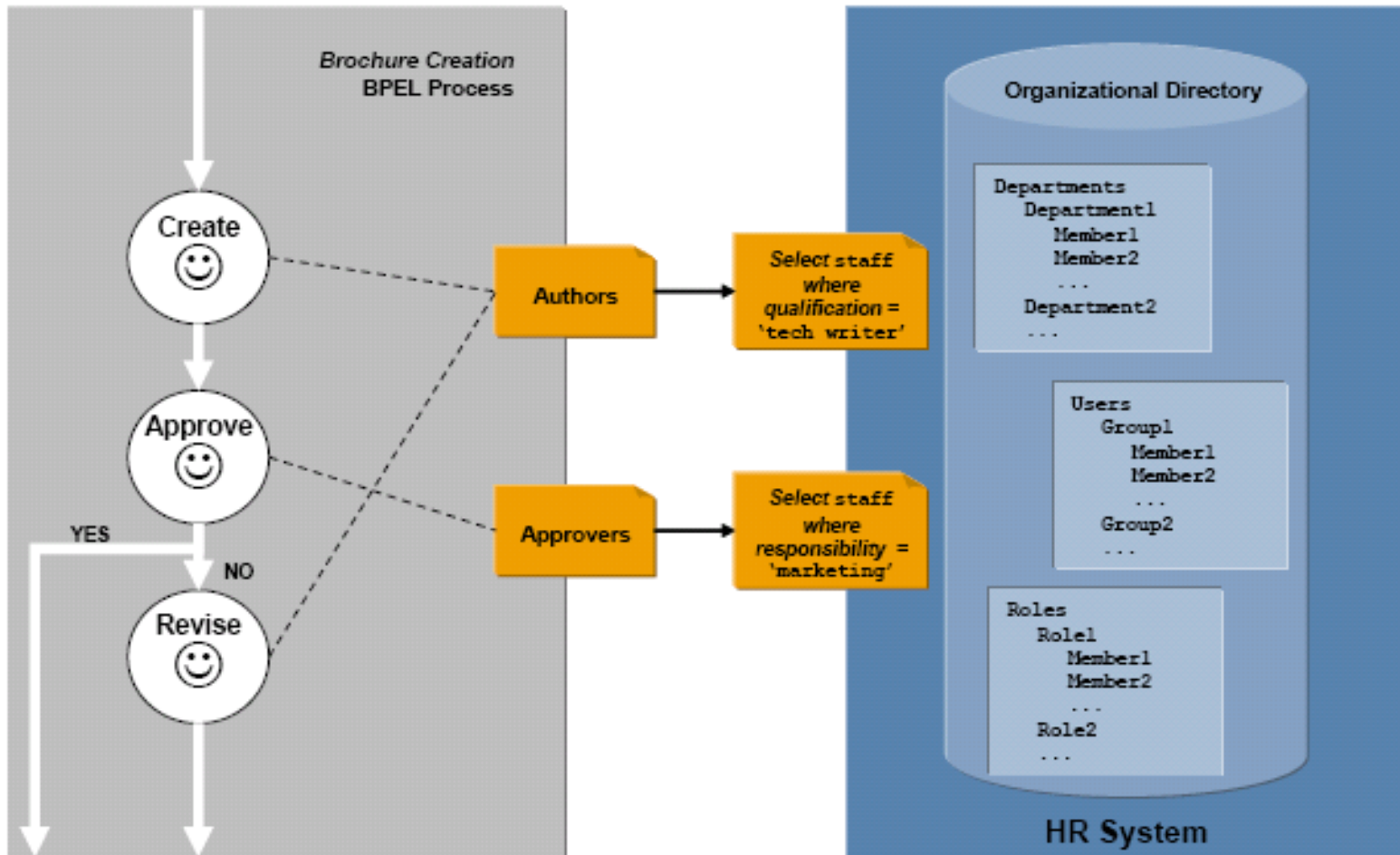
e"/

contrat WSDL du service invoqué

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://www.eclipse.org/tptp/choreography/2004/engine/Print"
  xmlns:tns="http://www.eclipse.org/tptp/choreography/2004/engine/Print"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:format="http://schemas.xmlsoap.org/wsdl/formatbinding/"
  xmlns:java="http://schemas.xmlsoap.org/wsdl/java/" >
<!-- engine printout port -->
  <message name="PrintMessage">
    <part name="value" type="xsd:string"/> </message>
  <portType name="Print">
    <operation name="print">
      <input message="tns:PrintMessage"/> </operation>
    </portType>
  <binding name="PrintPortWsifBinding" type="tns:Print">
    <java:binding/>
    <format:typeMapping encoding="Java" style="Java">
      <format:typeMap
typeName="xsd:string" formatType="java.lang.String"/>
    </format:typeMapping>
    <operation name="print">
      <java:operation methodName="print" parameterOrder="value"/>
    </operation>
  </binding>
  <service>
    <port name="JavaPrintPort" binding="tns:PrintPortWsifBinding">
      <java:address className="org.eclipse(...)EnginePrinterPort"/> </port>
    </service>
  <partnerLinkType name="printLink">
    <role name="printService" portType="tns:Print"/>
  </partnerLinkType>
</definitions>
```


BPEL 4 People

- Les humains doivent être pris en compte dans des interactions longues



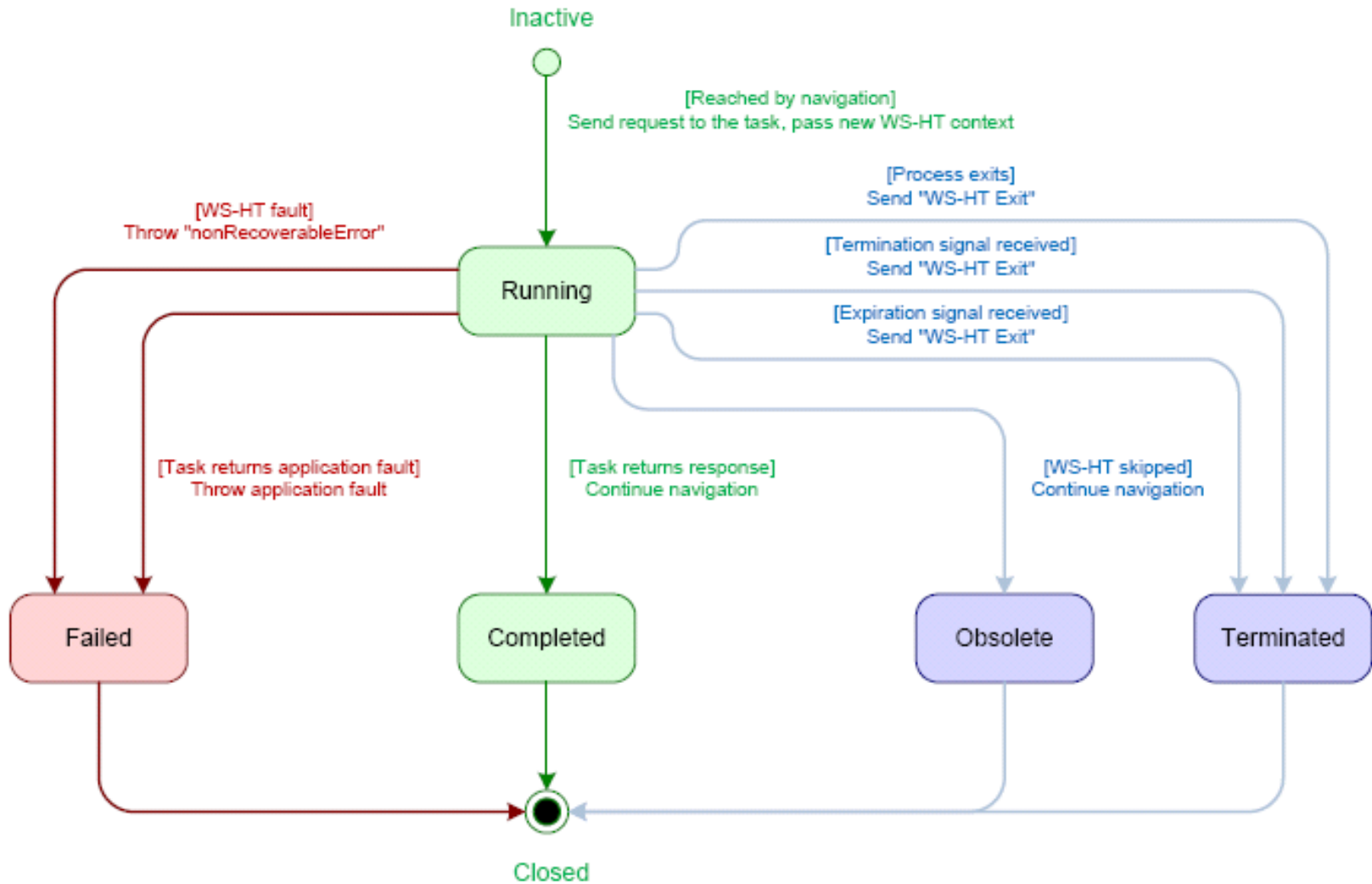
BPEL 4 People : vocabulaire

- **Business Administrator.** A person with an interest in the smooth running of all process instances of a particular process type.
- **Generic human role.** A descriptor identifying how a person interacts with the process. It includes the specific human roles such as potential owners, owners, and process initiator.
- **Inline task.** A task defined within a BPEL process either as part of a people activity or within the scope of a BPEL process so that it can be reused in different parts of the same BPEL process.
- **Owner.** A potential owner of a task becomes the owner when he or she claims the activity.
- **People activity.** A new BPEL activity that assumes that people are involved in the execution of the process activity.
- **People links.** A people link represents the group of people associated with a process or a people activity. It is evaluated at runtime using a people query.

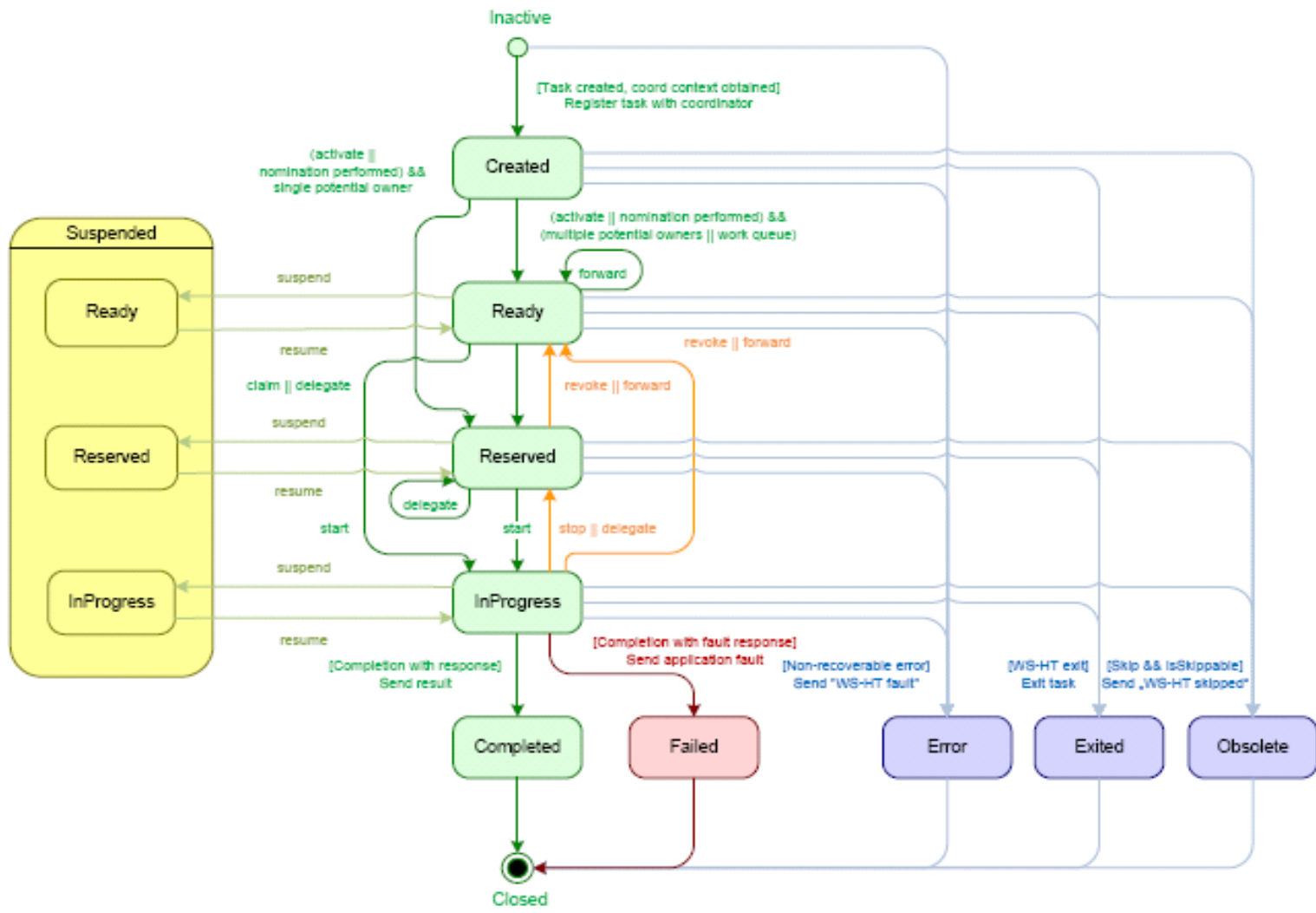
BPEL 4 People : vocabulaire

- **People query.** An expression based on an organizational model used to resolve users who take on the responsibility for a particular generic human role.
- **People resolution.** The act of identifying the people who take on the responsibility for a particular generic human role.
- **Potential owners.** People who are entitled to claim and complete an activity. This is one type of generic human role.
- **Process initiator.** A person that creates an instance of a process, either directly or indirectly. This is one type of generic human role.
- **Process stakeholder.** A person with an interest in the outcome of the process instance. This is one type of generic human role.
- **Standalone task.** A task defined outside the scope of a BPEL process, but it can be referenced and invoked from different BPEL processes.
- **Task.** An indivisible unit of work performed by a human being.

États d'une tâche



États et transitions des tâches humaines



Formes d'implémentation d'interactions entre tâches humaines et automatiques

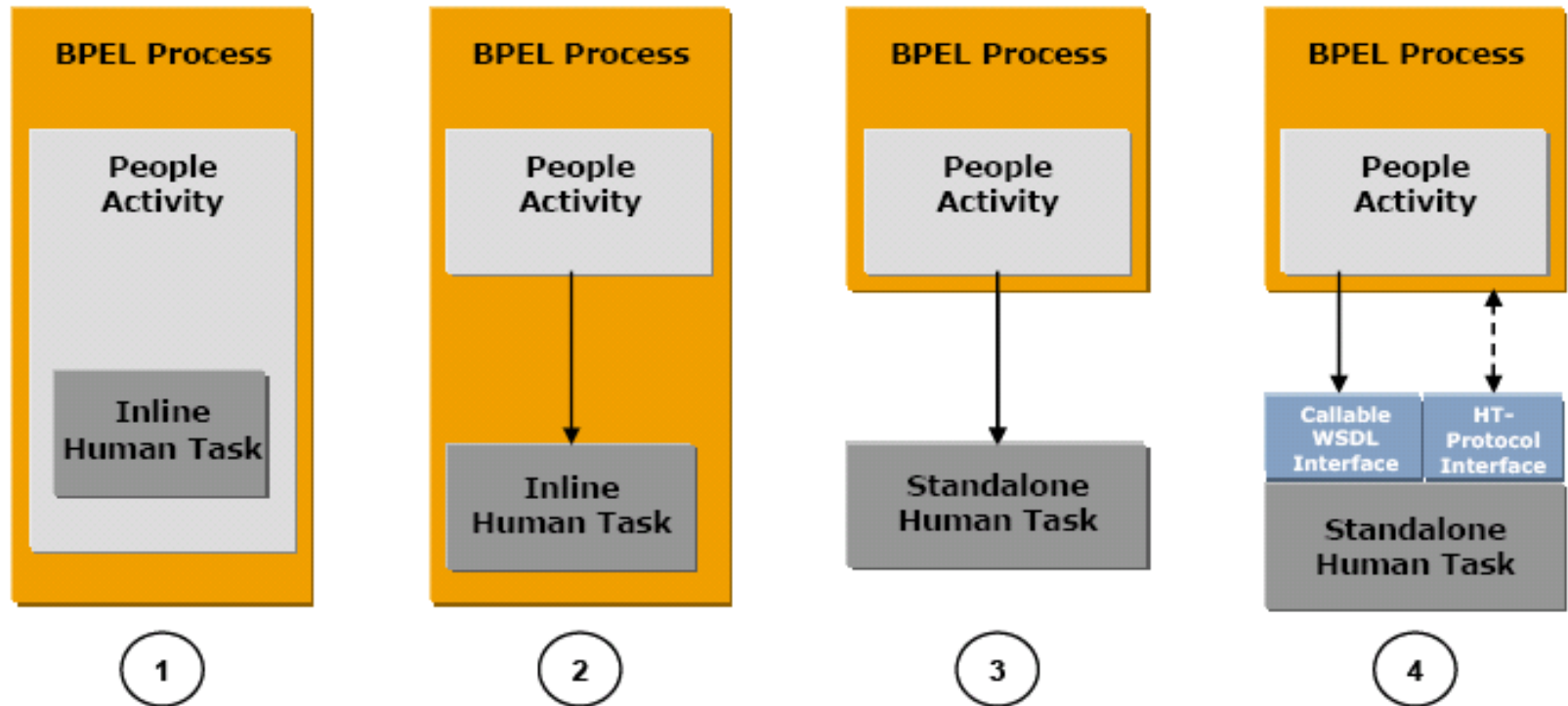


Figure 1: Constellations

Échanges de messages entre applications et humains

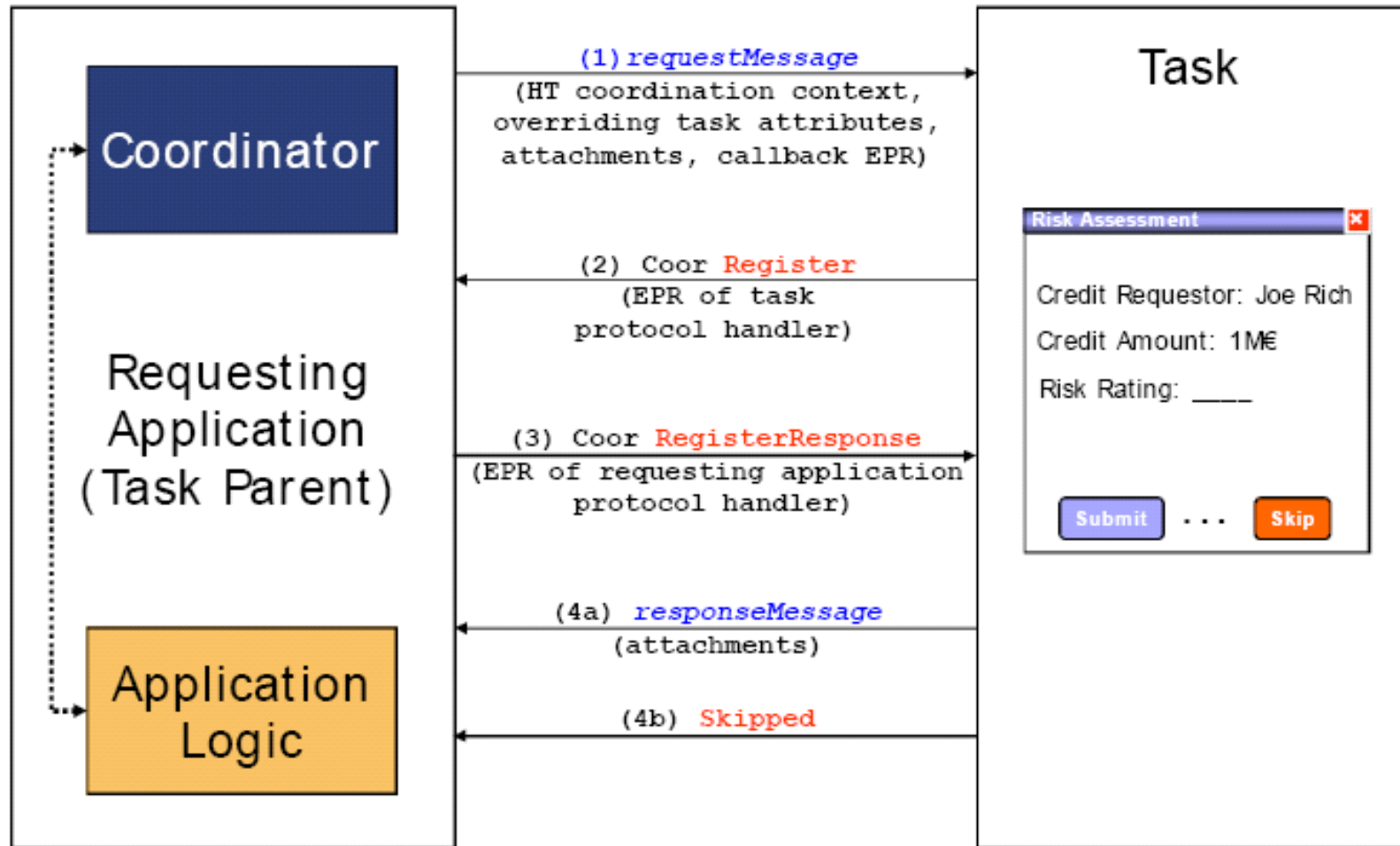


Figure 1: Message Exchange between Application and Human Task

- BPEL n'est qu'un nouvel avatar des multiples formalisation de processus opérationnel,
 - (Petri, Grafset etc...)
 - Adapté à la problématique de l'appel des web services
 - Nécessaire et suffisant pour les adeptes du web commercial,
 - Exigeant un minimum de culture informatique.
 - Nécessitant une modalité d'expression graphique
- BPMN (Business Process Model Notation)
 - Est une notation d'expression graphique homologue d'une expression BPEL
 - Se veut intuitif pour être adapté à la culture des adeptes du commerce électronique

Core Set of BPMN Elements

Flow Objects

Events



Activities



Gateways



Connecting Object

Sequence Flow



Message Flow

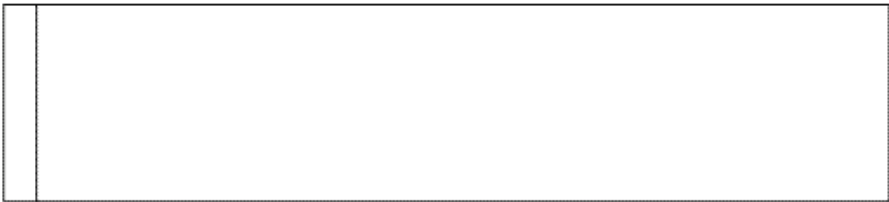


Association

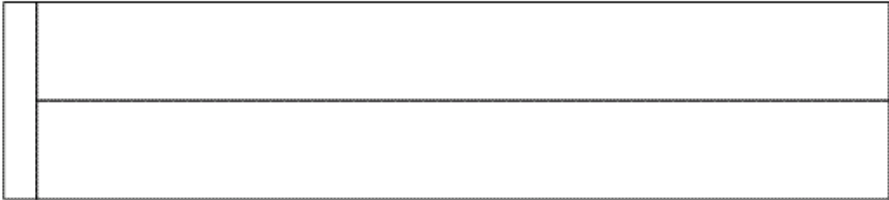


Swimlanes

Pool



Lanes (within a Pool)



Artifacts

Data Object



Name
[State]

Text Annotation

Text Annotation Allows a Modeler to provide additional Information

Group

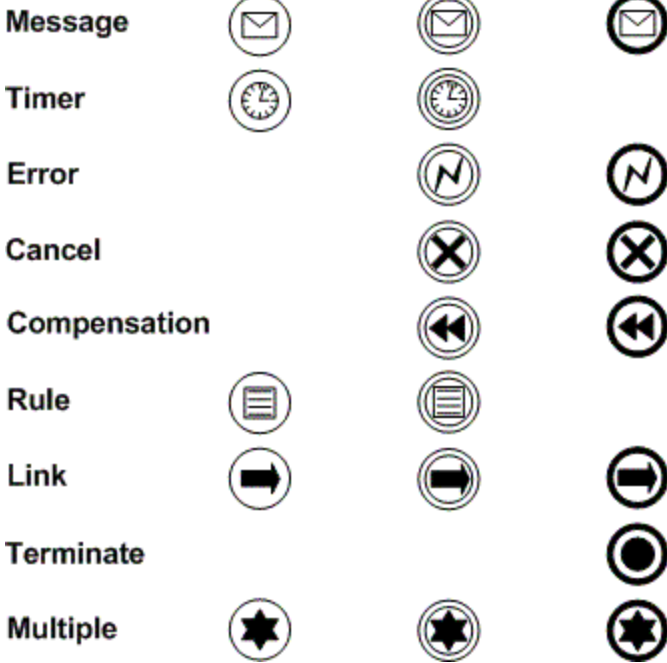


BPMN : Événements et activités

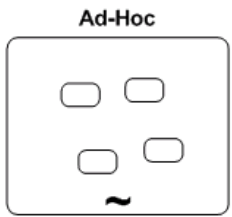
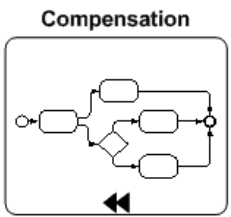
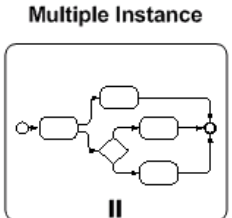
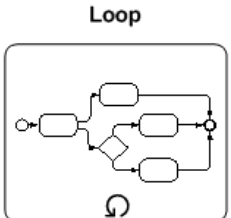
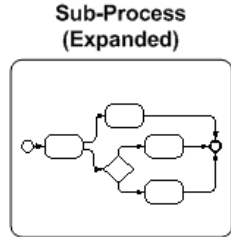
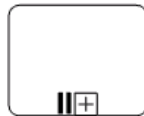
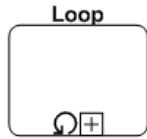
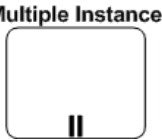
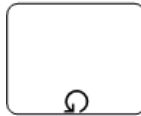
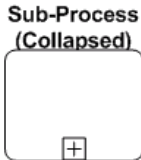
Events



Event Types

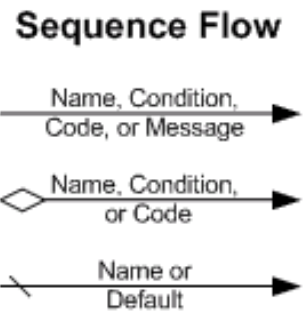
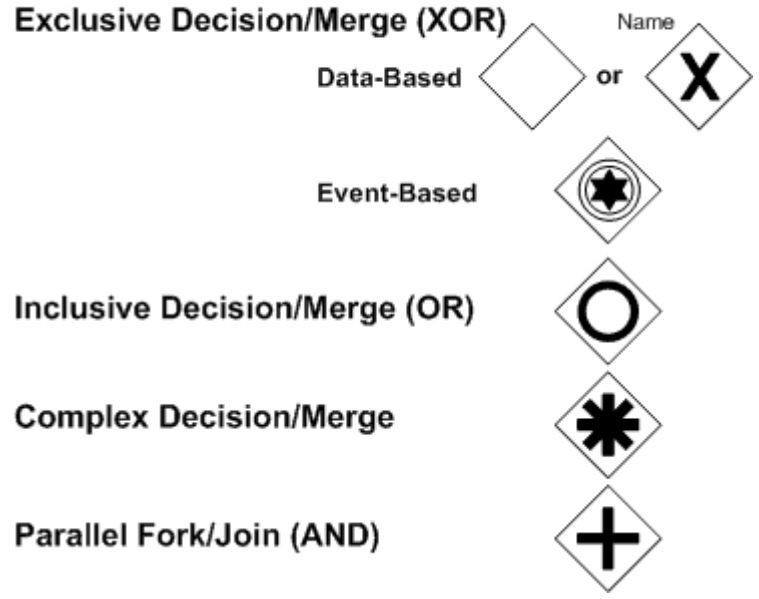


Activities

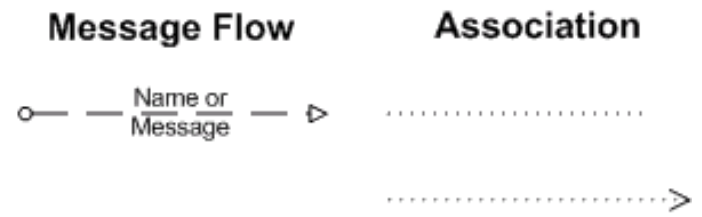


BPMN : Tests et connexions

Gateways



Connections



BPMN : Artefacts et « lignes de nage »

Artefacts

Data Object



Name
[State]

Text Annotation

 Add Text Here

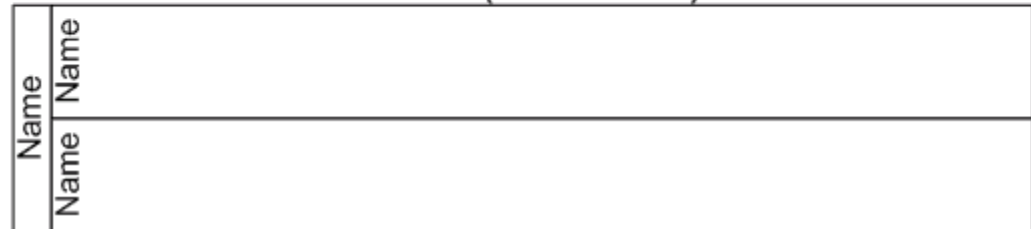
Group



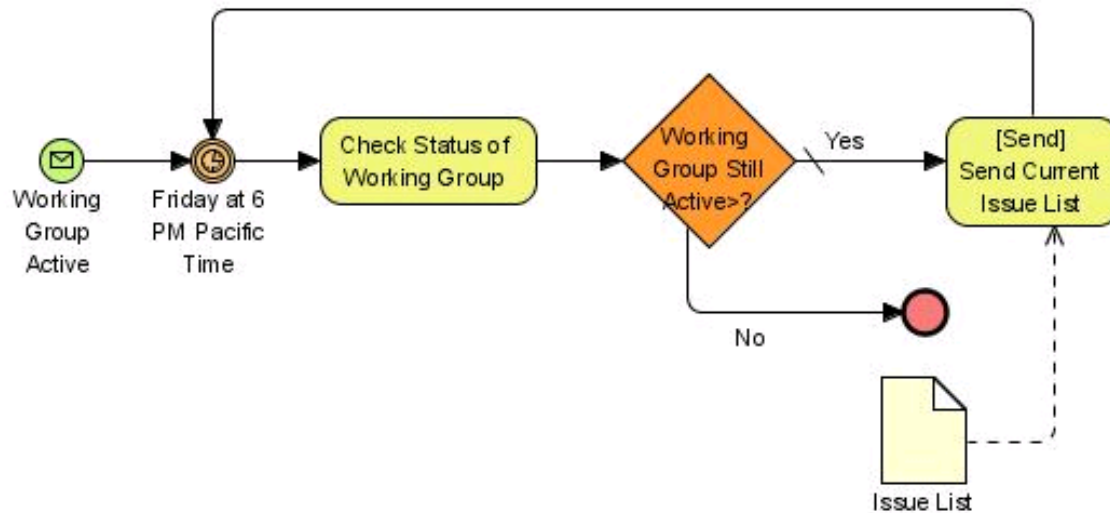
Swimlanes Pool



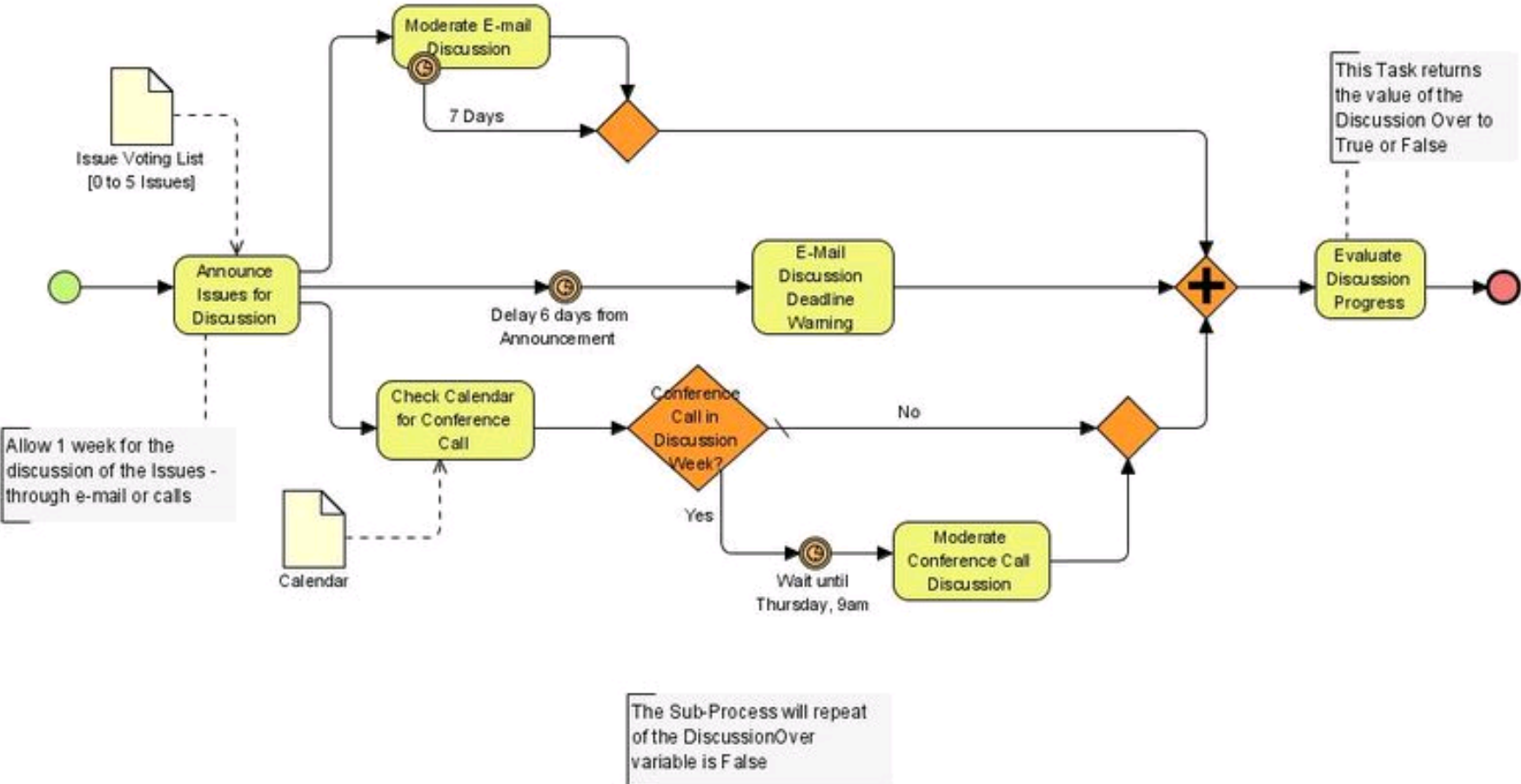
Lanes (within a Pool)



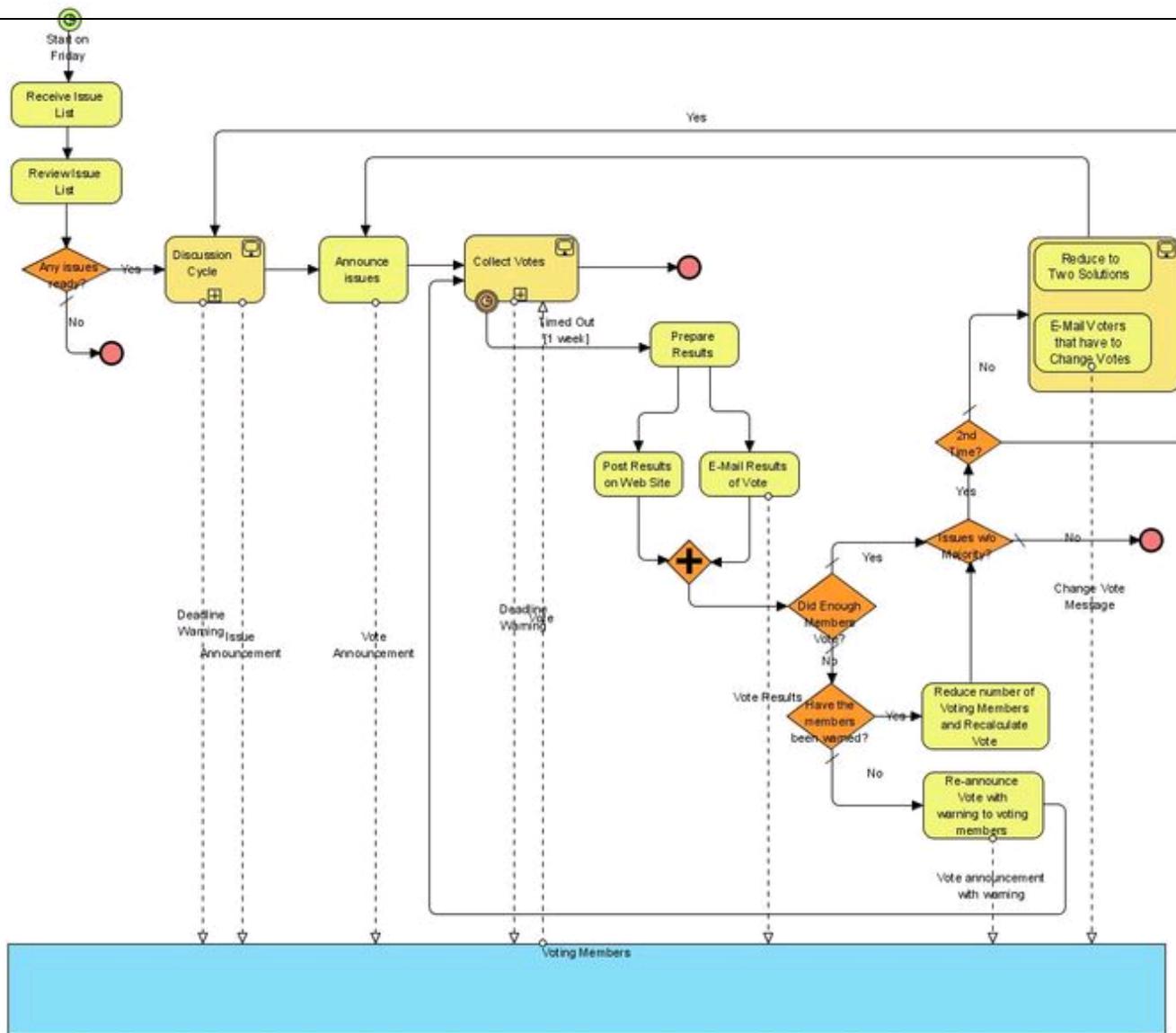
- Exemple de processus simple de rappel, par un web service, des questions à traiter par un groupe de travail



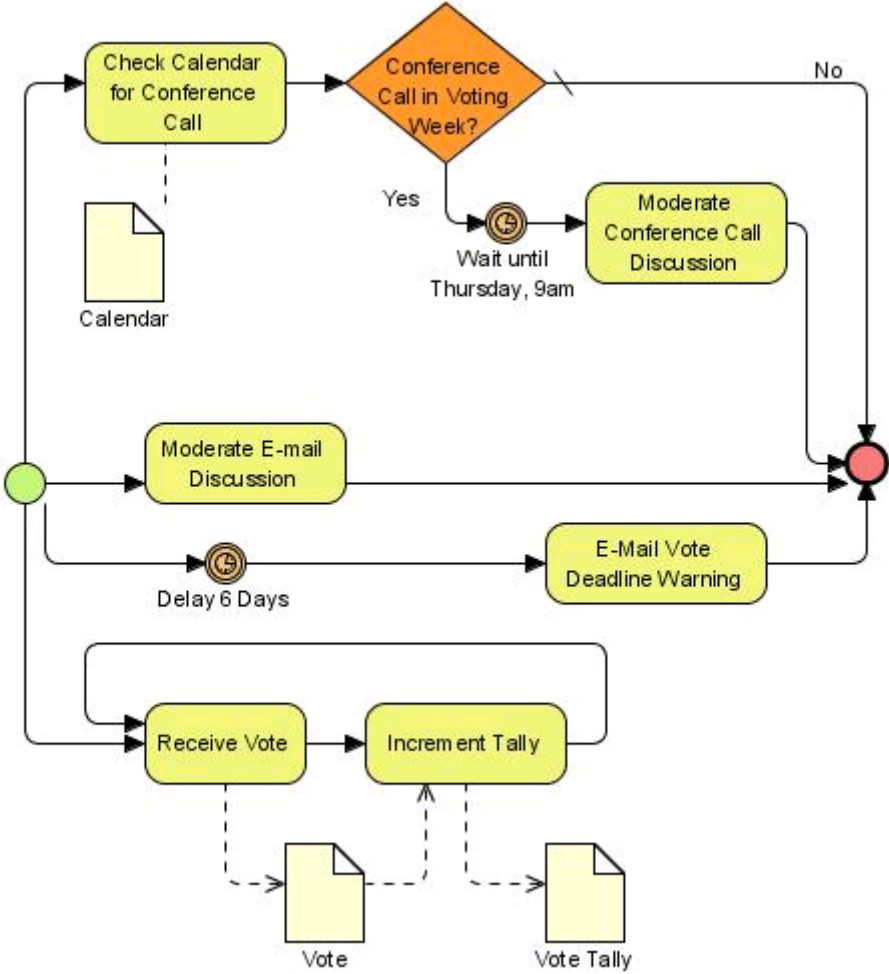
BPMN : discussion



BPMN : vote par e-mail



BPMN : dépouillement des votes



Fin du module